

Research Article

An Automatic Density Clustering Segmentation Method for Laser Scanning Point Cloud Data of Buildings

Jianghong Zhao ¹, Yan Dong ¹, Siyu Ma ¹, Huajun Liu ², Shuangfeng Wei ¹,
Ruiju Zhang ¹ and Xi Chen ³

¹School of Geoinformatics and Urban Information, Beijing University of Civil Engineering and Architecture, Beijing, China

²Huanqiujiingwei Surveying and Mapping Engineering (Beijing) Co., Ltd., Beijing, China

³Beijing Research Base for Architectural Culture, Beijing University of Civil Engineering and Architecture, Beijing, China

Correspondence should be addressed to Jianghong Zhao; zhaojiangh@bucea.edu.cn

Received 22 March 2019; Revised 29 May 2019; Accepted 3 June 2019; Published 7 July 2019

Academic Editor: Ana C. Teodoro

Copyright © 2019 Jianghong Zhao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Segmentation is an important step in point cloud data feature extraction and three-dimensional modelling. Currently, it is also a challenging problem in point cloud processing. There are some disadvantages of the DBSCAN method, such as requiring the manual definition of parameters and low efficiency when it is used for large amounts of calculation. This paper proposes the AQ-DBSCAN algorithm, which is a density clustering segmentation method combined with Gaussian mapping. The algorithm improves upon the DBSCAN algorithm by solving the problem of automatic estimation of the parameter neighborhood radius. The improved algorithm can carry out density clustering processing quickly by reducing the amount of computation required.

1. Introduction

Light Detection and Ranging (LIDAR) is an important means of obtaining building data. Because the point cloud data obtained by LIDAR are voluminous and discrete, and usually have significant amounts of noise data, automatic semantic segmentation and feature extraction have become a bottleneck of current research and are academic hotspots at the forefront of this field.

At present, the commonly used segmentation methods include the region-growing segmentation method [1–4], model fitting segmentation method [5, 6], mixed segmentation method [6, 7], and clustering-based segmentation method [8–11].

The region-growing segmentation method is simple and efficient, but it is easy to segment excessively due to the interference of noise points [12, 13]. The model fitting segmentation method is robust against interference; it has a significant disadvantage in that it can only detect basic geometry and cannot detect complex surfaces [6, 14–18]. Additionally, due to the vast number of matching iterative operations, its efficiency is low. The mixed segmentation method integrates

multiple segmentation methods and is generally designed for specific point cloud data [19, 20]. In order to obtain good applicability, the algorithm inherits advantages and abandons or weakens defects.

There are three main kinds of clustering-based segmentation methods, including hierarchical methods, partitioning methods, and density-based methods [21, 22].

Hierarchical methods decompose a collection hierarchically until certain conditions are met. Because of the irreversibility of hierarchical methods, errors cannot be corrected in the process of decomposition or clustering.

The k-means algorithm is one of the most common partitioning methods. The k-means algorithm uses value k as a parameter to divide n objects into k clusters so that the objects within the same clusters have higher similarity, while the similarity between different clusters is low. Because of its simplicity and speed, it can be used for very large datasets. However, the k-means algorithm is easily affected by the initial value, and the value k must be given in advance.

The key idea of density-based clustering is to judge whether the sample points belong to a cluster by whether they are closely related. In the view of the density clustering

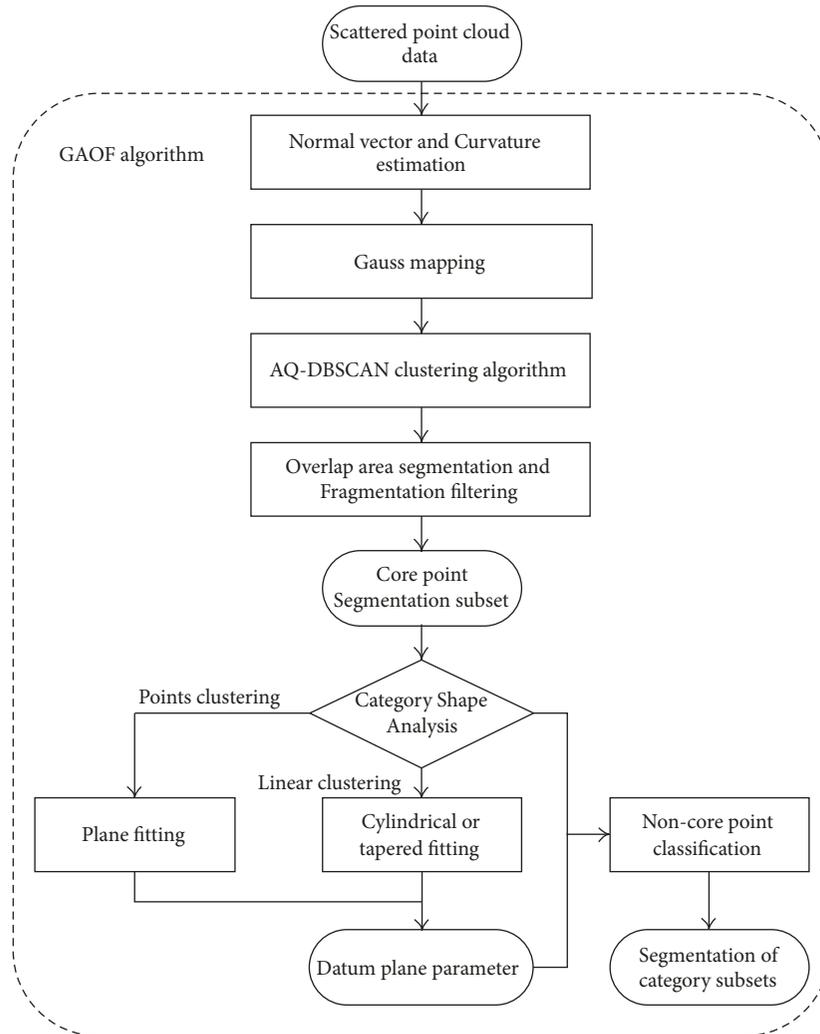


FIGURE 1: The segmentation method flow chart.

method, the cluster is a high-density object region separated by a low-density region in data space, and the data in the sparse data region is considered as noise data. This algorithm sets a certain threshold, and as long as the density of the adjacent area of a certain point exceeds the threshold, the cluster can proceed. This method can find any shape of cluster and can filter the “noise” data. Finding the cluster shape is the greatest advantage of the algorithm, and it is widely used in the practical algorithm. A typical algorithm for this method is the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm [23]. The main disadvantage of this type of method is that it is more sensitive to user-defined density parameters, that different thresholds for the results of clustering are relatively large and that the threshold must be manually set. In addition, many iterative computations are needed in the clustering process, which leads to low efficiency. The FDBSCAN algorithm proposed by Zhou [24] is an improvement on the DBSCAN algorithm. This algorithm proposed a method of determining the number of representative points according to the clustering dimension, which can reduce the number of iterations. However, it still

requires setting of the neighborhood radius manually and cannot be fully automated.

Focusing on the advantages and disadvantages of the above methods, this paper proposes AQ-DBSCAN, an algorithm that improves upon the DBSCAN algorithm by including an automatic segmentation method for point cloud data obtained from buildings. The method used in the AQ-DBSCAN algorithm is designed to segment a point cloud mapped on a Gaussian sphere. Based on the minimum distance, maximum distance, and number of points in different neighborhood ranges on the Gaussian sphere, the value of space sphere radius σ is obtained automatically. The AQ-DBSCAN algorithm solves the problem of automatic estimation of the neighborhood radius in the DBSCAN algorithm and reduces the amount of computation in density clustering. The specific segmentation process is shown in Figure 1.

Through Gaussian mapping, the data can be reduced from three dimensions to two dimensions and the influence of interference points on subsequent segmentation can be removed. The clustering can also get shape features, which is convenient for subsequent feature extraction.

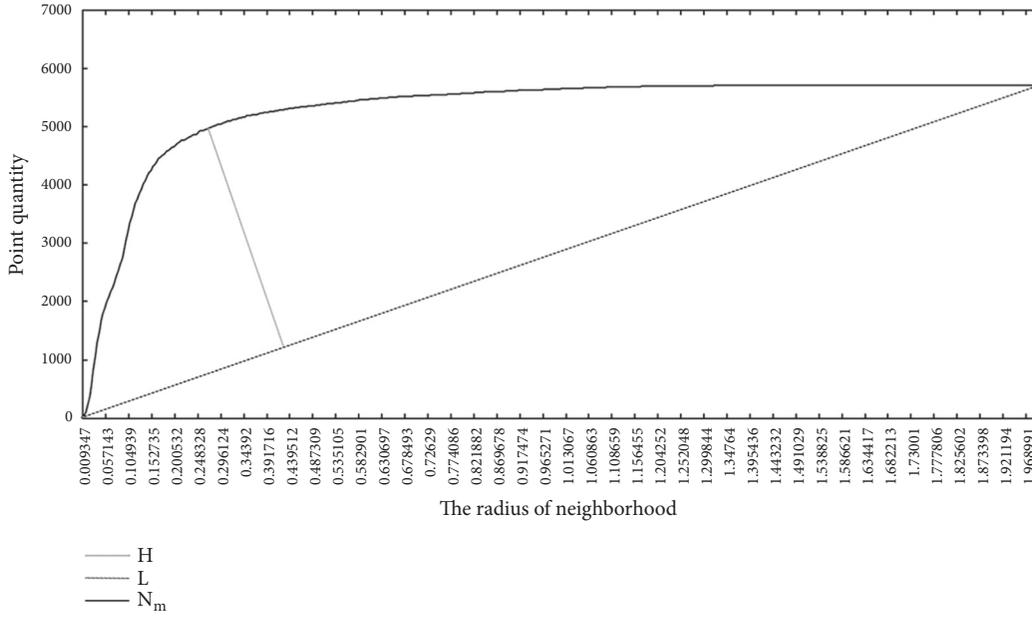


FIGURE 2: The N_m graph.

2. DBSCAN Algorithm

DBSCAN is a classical algorithm for density clustering. The DBSCAN algorithm requires two parameters in advance. The first parameter is the neighborhood radius σ . The second parameter is MinPts, which is the threshold of the number of points in the core point neighborhood. Clusters can be established by directly and iteratively searching density-reachable points. Ester [23] gives the selection method of these two parameters:

- (1) Specify MinPts as 4; then for any $x \in X$, calculate the maximum distance $D(x)$ of the four objects closest to x_0 , and then traverse all the points in X to get the set $D = \{D(x) \mid x \in X\}$.
- (2) Sort D , and when drawing the curve, the abscissa is the number of points, and the ordinate is the distance.
- (3) Calculate and find the inflection point on the curve of the rapid rise, and the distance of the inflection point is the neighborhood radius σ .

The geometric shapes of ancient building elements mainly include planes, cylinders, and the like, which are mapped to Gaussian balls for the point or line shape. The DBSCAN algorithm can identify any shape of cluster and has strong applicability for the initial segmentation of the ancient building Gaussian map point cloud. However, as the DBSCAN algorithm performs a neighborhood query and calculation for each point, its operation efficiency is low, having a time complexity of $O(n^2)$. For a point cloud of ancient buildings with a large amount of data, the DBSCAN algorithm is particularly inefficient. The neighborhood radius σ is usually determined by hand and is obviously not suitable for automatic point cloud segmentation.

3. AQ-DBSCAN: An Improved Automatic Clustering Algorithm

3.1. Automatic Parameter Estimation. The DBSCAN parameter estimation proposed in this paper is an artificial judgment method, which cannot satisfy the requirement of automatic point cloud segmentation. According to the DBSCAN algorithm, MinPts is set to 4. $G(x_i)$ is an element in $G(X)$, which is mapped on the Gauss sphere by the X -point set. Search the K neighborhood of the original point x_i corresponding to $G(X)$ in X and then find the MinPtsth point closest to the $G(x_i)$ after the point set of the K neighborhood is mapped to the Gaussian sphere. Let the distance be D_{MinPts} , calculating the D_{MinPts} for each point, and the total number of corresponding points with Δ as the increment unit interval is counted as follows:

$$N_m = \Delta N G(X), \quad 0 < D_{\text{MinPts}} \leq \Delta \times m \quad (1)$$

The graph takes Δ as the horizontal axis and N_m as the vertical axis. As m increases, the curve of N_m gradually flattens out. The N_m graph of the point cloud data is generally shown in Figure 2.

In Figure 2, because the plane and the cylinder have certain differences in the D_{MinPts} on the Gaussian sphere, the N_m curve exhibits a slight wave fluctuation during the upward extension. The start and the end points of the connection curve form a straight line L and are then used to calculate the distance H of each point on the curve from the straight line L as well as the largest statistical point of H . The x -value of the point is the value of the parameter σ . In this way, the automatic estimation of the parameter σ is achieved.

3.2. Fast Clustering. In the growth of a class, the DBSCAN algorithm needs to determine whether each point in a

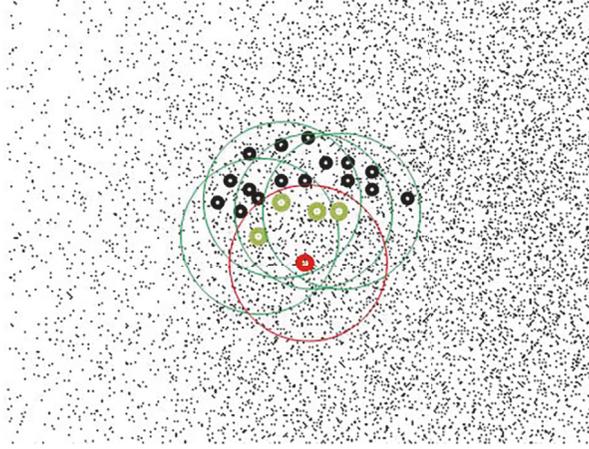


FIGURE 3: The influence of distribution of representative points on clustering effect.

neighborhood is a core point, which is accomplished by calculating the number of points in the neighborhood of $\sigma(x_i)$. Although the K-D tree index can be established on the point cloud data after Gaussian mapping to increase the judgment speed, the operation efficiency of the algorithm is still low.

The FDBSCAN algorithm proposed by Zhou [24] is an improved version of DBSCAN. The FDBSCAN algorithm proposes selecting the number of representative points (rather than all neighborhood points) in the neighborhood to replace the class growth. The number of representative points is related to the dimension of space; i.e., the number of representative points of n -dimensional space is $2n$.

Figure 3 shows the effect of the divergence of four representative points on the clustering expansion effect of two-dimensional space. In the case where the divergence of the representative point is not good, there is a case where the density of a point and x_i is reachable, but it can only be reached through the core point density, which is not selected as the representative point. In Figure 3, the red dot is x_i , the red circle is the region of $\sigma(x_i)$, the green point is the representative point selected from $\Sigma(x_i)$, the green circle is the neighborhood area of the representative point, and the black points are representative points which are further selected from the neighborhood of the representative point to make the cluster extension. Because the representative points are concentrated in the upper region of the neighborhood, the expansion direction of the clustering is developed upward, and the lower points cannot be classified into the same category.

In view of the above problems, on one hand, we need to combine the points and the relevant subclasses of these points in follow-up work, and on the other hand, we should take into account the rapidity of the algorithm and the diffusion of the representative points in the selection of representative points. In [24], a selection algorithm of two representative points is proposed, but neither the efficiency of algorithm nor the divergence of representative points is suitable for the dense point cloud data on Gaussian spheres. Compared

with the FDBSCAN algorithm, AQ-DBSCAN reduces the amount of computation in the search algorithm with the two-dimensional characteristics of data on the Gaussian spherical surface and sets the number of representative points as 4. Additionally, this algorithm presents a means for selecting representative points that are suitable for dense point cloud data, which enhances the selection efficiency of representative points and guarantees the diffusion of representative points. The selection method is described as follows:

Set x_0 as an object in point set X , $0 < \varepsilon < \sigma$; then

$$\varepsilon(x_0) = \{x \in \sigma(x_0) \mid \varepsilon < D(x, x_0) \leq \sigma\} \quad (2)$$

$\varepsilon(x_0)$ is a candidate set of representative points in an x_0 neighborhood. $\varepsilon(x_0)$ is the annular region between ε and σ centered on x_0 . Selecting a representative point in the peripheral area of $\sigma(x_0)$ helps to enhance the diffusion of the representative points and reduces the frequency of neighborhood queries when the class expands. The closer the value of ε is to the value of σ , the fewer the number of candidate points is. In this case, the search efficiency of the representative points will be higher, but may also lead to a poor divergence of points. Considering the factors of diffusion and high efficiency synthetically, the ε value of this algorithm is $\sigma - \sigma/4$. When the representative point is selected, the point furthest from x_0 is searched as the representative point, and the selection of the subsequent representative point is a three-iteration of search in $\varepsilon(x_0)$.

Set the represented set of points to P , and set the representative point to be searched to p_k ; then,

$$p_k = \max_{x \in \sigma(x_0)} D(x, P) \quad (3)$$

Among them,

$$D(x, P) = \min_{p_i \in P} \|x - p_i\|_2 \quad (4)$$

as shown in Figure 4.

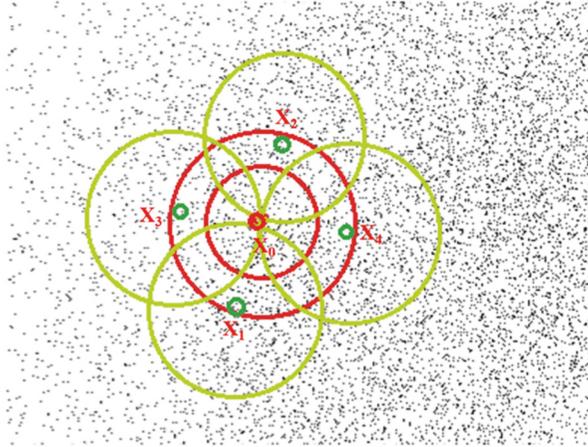


FIGURE 4: Selection and diffusion of AQ-DBSCAN representative points.

There are four steps used to find the representative point in the area of the red ring (located between the p and k ring regions). First, find the point X_1 , which is the furthest point from the search point X_0 . Second, find the furthest point X_2 with X_1 . The third step is to find the smallest maximum-distance point with X_1 and X_2 . Finally, find the maximum distance point with X_1 , X_2 , and X_3 .

3.3. Practical Comparison among AQ-DBSCAN, DBSCAN, and FDSCAN. Compared with DBSCAN and FDBSCAN, there are two improvements made by AQ-DBSCAN. First, the automatic estimation of σ is provided with the precondition of the given MinPts parameter. Second, a faster density-based spatial clustering algorithm is achieved. In this paper, all the experiments are completed by a notebook computer with 1.2GHz dual-core CPU and 1 GB memory. The experiment data is point cloud data, which was collected using a terrestrial laser scanner from the Gate and the Hall of Supreme Harmony in the Forbidden City in Beijing, China. The scanner model used was a ScanStation2, shown in Figure 5, and is a pulsed scanner with a scanning speed of 50,000 dps. The maximum scanning angle of a single station is $360^\circ \times 270^\circ$, the maximum distance of a single station is 300 meters, and the accuracy of the scanner is better than 6 mm.

Figure 6(a) shows the columns, which are located outside the Hall of Supreme Harmony, and the number of column samples of point cloud data is 141,999. Figure 6(b) is the corresponding Gauss map. Figure 6(c) includes the N_m line graph counted automatically by the AQ-DBSCAN algorithm and the estimated value of σ (0.067831). The DBSCAN and FDBSCAN algorithms mainly rely on the expert's individual judgment of the shape of the N_m curve and the change of the extension trend in parameter estimation. According to the figures, the AQ-DBSCAN algorithm gives the σ value exactly at the inflection point of the N_m curve, which is consistent with the result of manual judgment. Figure 6(d) is the Clustering Effect Diagram on the Gaussian sphere. The cylinder and some appendages are clustered into one class because they overlap on the Gaussian sphere, which



FIGURE 5: ScanStation2.

will be segmented after the analysis of the overlapping zone. Figure 6(e) is the corresponding Clustering Effect Diagram to a spacing surface.

Figure 7 illustrates parameter estimation and clustering effect of the beam above the Hall of Supreme Harmony. Figure 7(a) shows the beam above the Hall of Supreme Harmony, and the number of the beam samples of point cloud data is 642,984. Figure 7(b) is the corresponding Gauss map. Figure 7(c) includes the N_m line graph counted automatically by the AQ-DBSCAN algorithm and the estimated value of σ (0.011236). According to the figures, the AQ-DBSCAN algorithm gives the σ value exactly at the inflection point of the N_m curve, which is consistent with the result of manual judgment. Figure 7(d) is the clustering effect diagram on the Gaussian sphere. Figure 7(e) is the corresponding Clustering Effect Diagram to a spacing surface. Based on the data, the AQ-DBSCAN cluster separates the front and side of the beam.

Figure 8(a) shows part of the data of the Hall of Supreme Harmony, including a column and the beams associated with

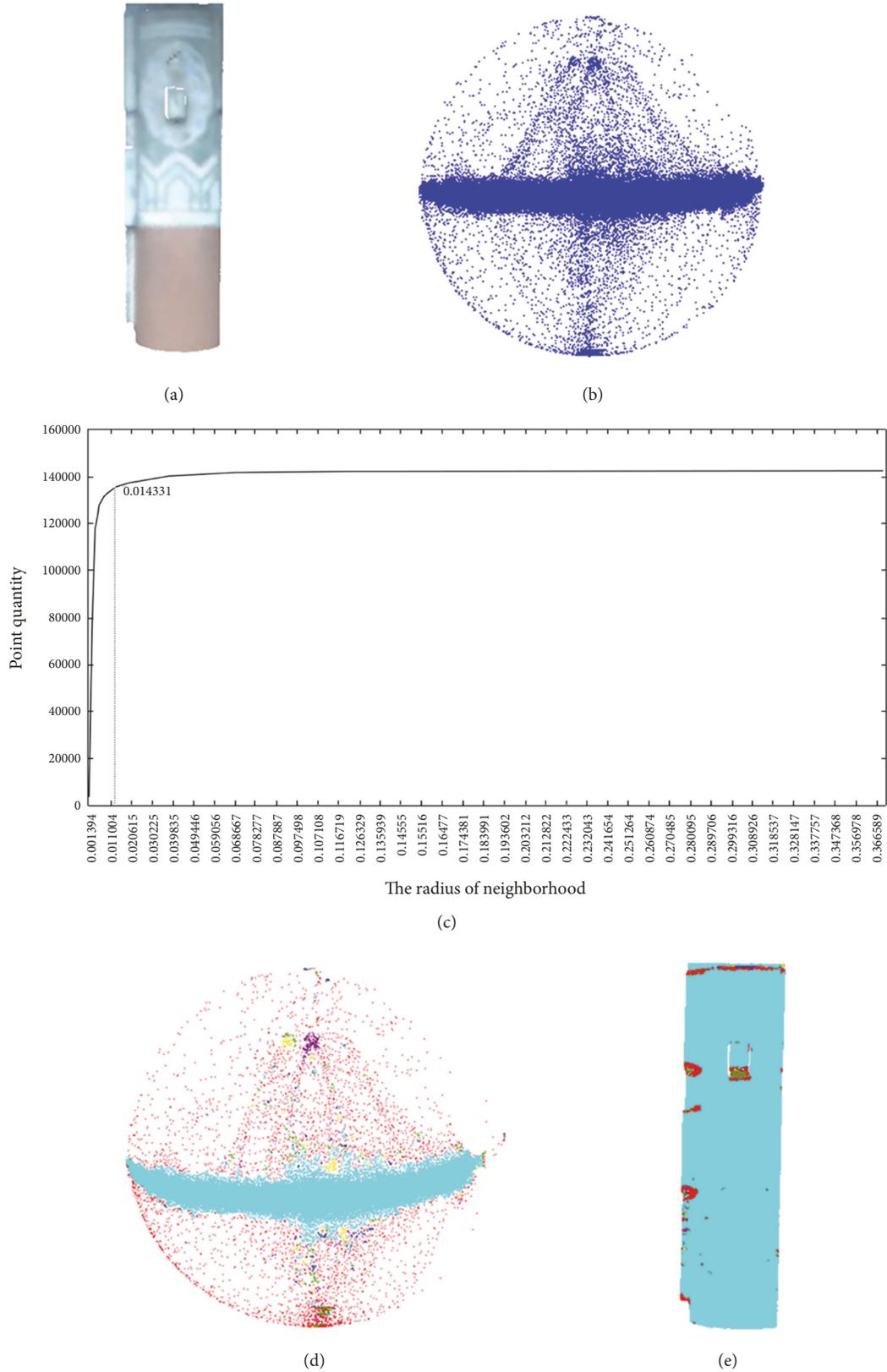


FIGURE 6: Parameter estimation and clustering effect of the columns located outside the Hall of Supreme Harmony. (a) Original point cloud. (b) Gauss map. (c) N_m line graph and the automatically estimated value of σ . (d) Clustering Effect Diagram on the Gaussian sphere. (e) AQ-DBSCAN Clustering Effect Diagram (space view).

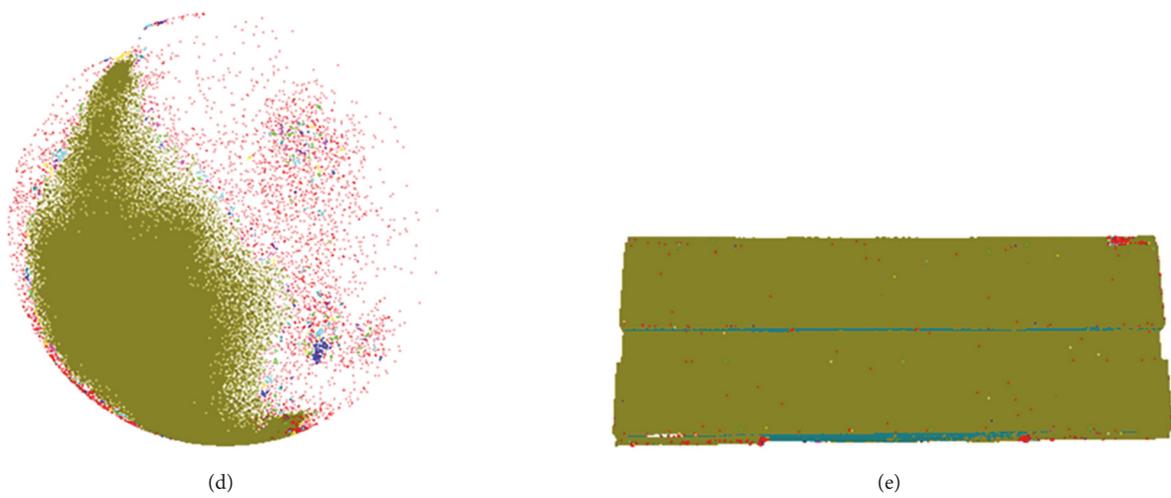
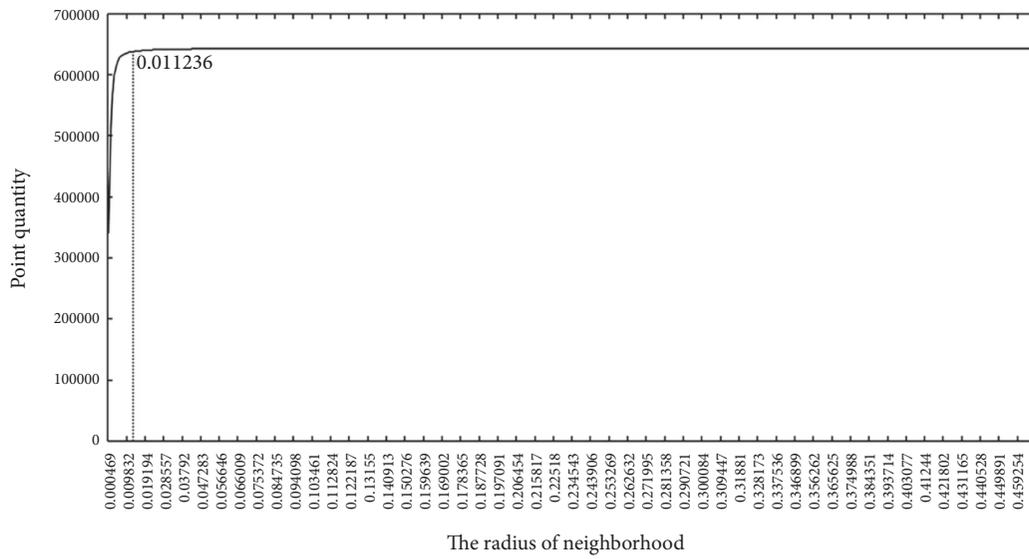
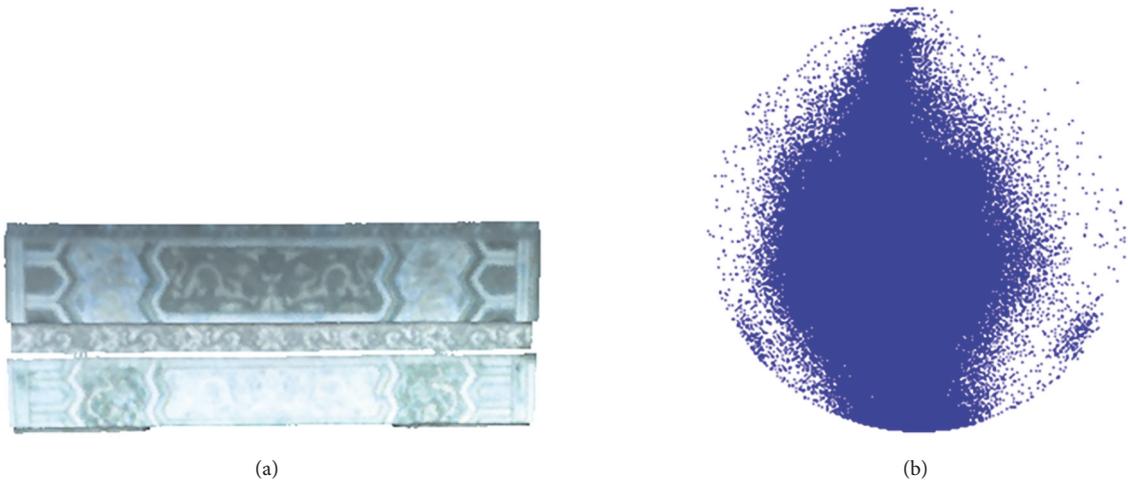


FIGURE 7: Parameter estimation and clustering effect of the beam above the Hall of Supreme Harmony. (a) Original point cloud. (b) Gauss Map. (c) N_m line graph and the automatically estimated value of σ . (d) Clustering Effect Diagram on the Gaussian sphere. (e) AQ-DBSCAN Clustering Effect Diagram (space view).

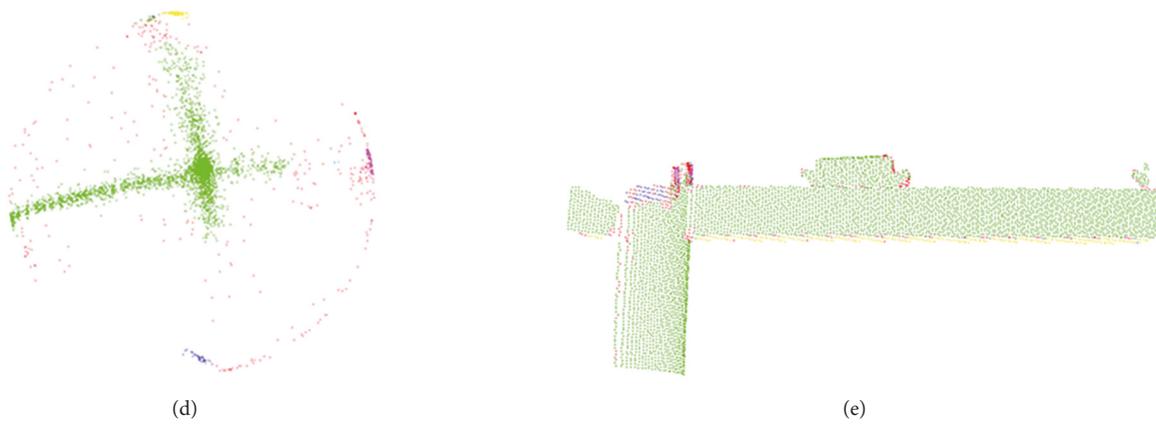
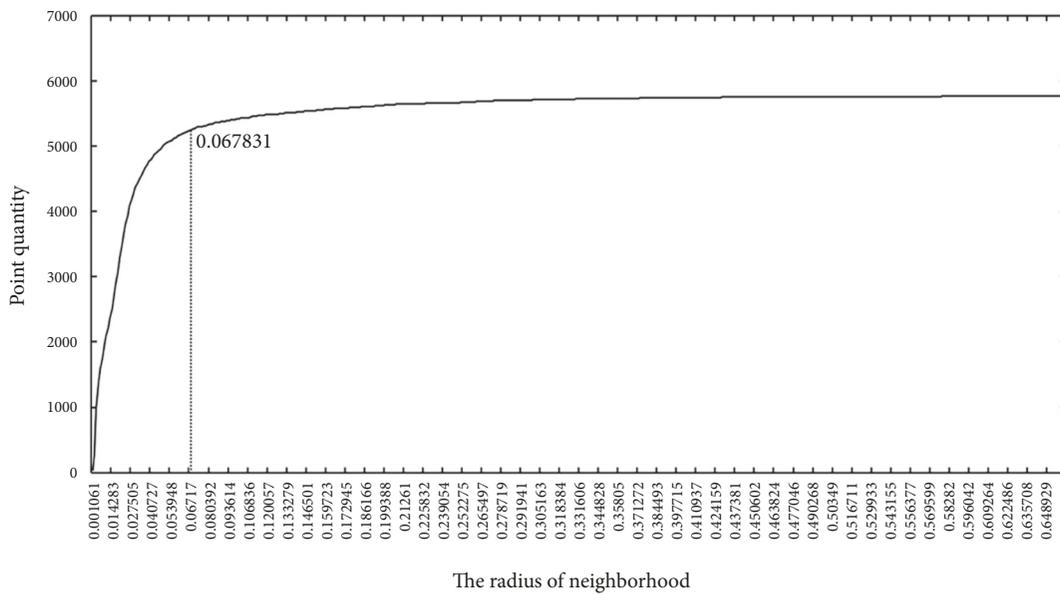
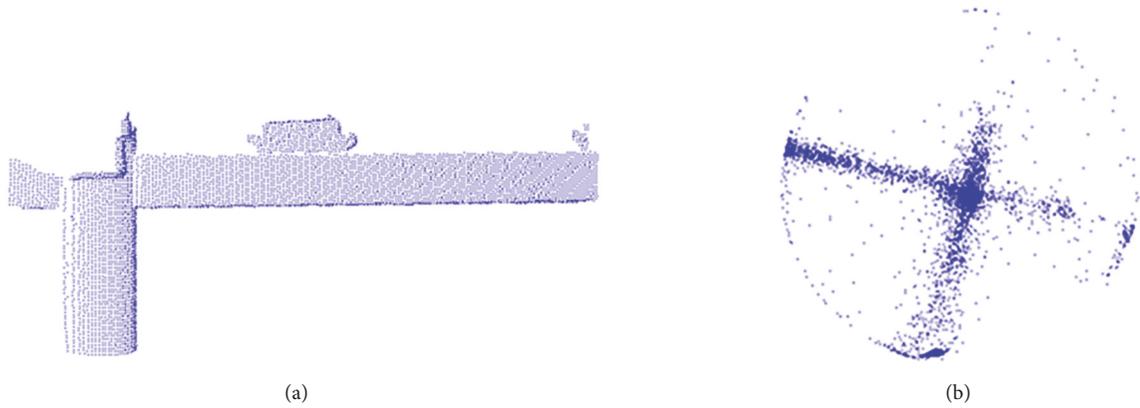


FIGURE 8: Parameter estimation and clustering effect of the part of the data of the Hall of Supreme Harmony. (a) Original point cloud. (b) Gauss Map. (c) N_m line graph and the automatically estimated value of σ . (d) Clustering Effect Diagram on the Gaussian sphere. (e) AQ-DBSCAN Clustering Effect Diagram (space view).

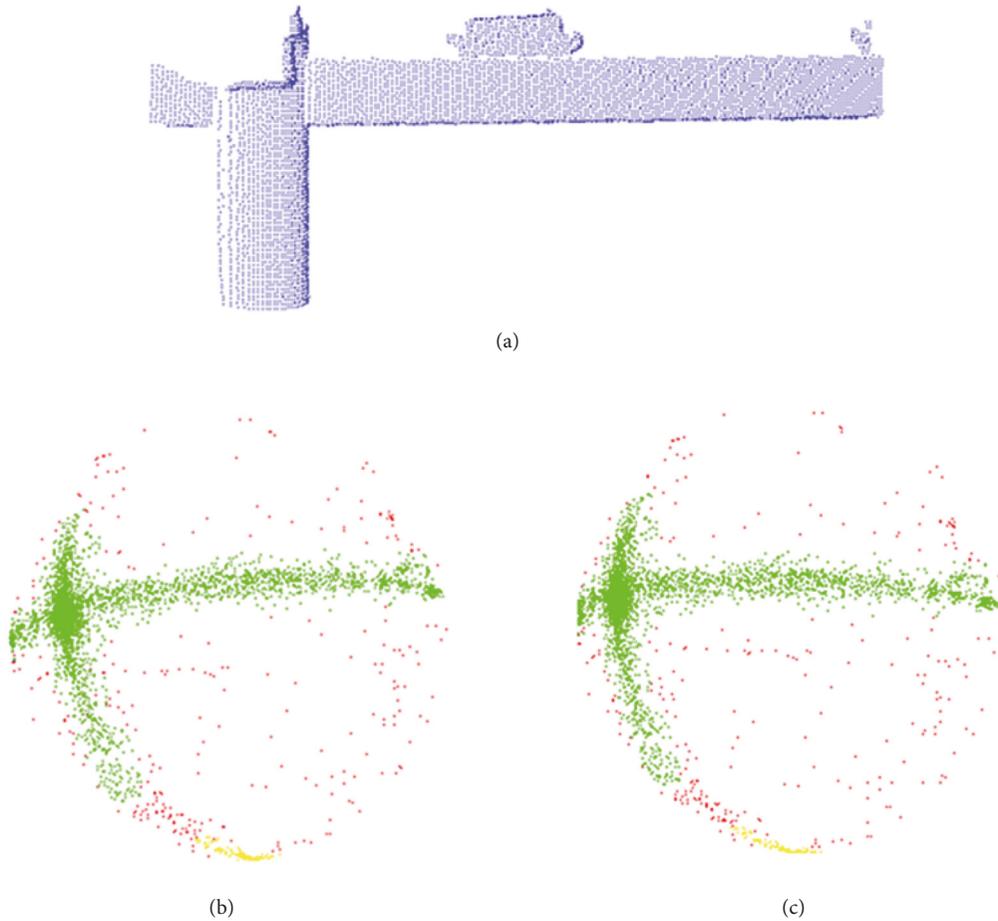


FIGURE 9: Comparison of the effects of DBSCAN clustering and AQ-DBSCAN clustering (1st). (a) Original point cloud. (b) DBSCAN Clustering Effect Diagram. (c) AQ-DBSCAN Clustering Effect Diagram.

it, and the number of the samples of point cloud data is 5,771. Figure 8(b) is the corresponding Gauss map. Figure 8(c) includes the N_m line graph counted automatically by the AQ-DBSCAN algorithm and the estimated value of σ (0.014331). According to the figures, the AQ-DBSCAN algorithm gives the σ value exactly at the inflection point of the N_m curve, which is consistent with the result of manual judgment. Figure 8(d) is the Clustering Effect Diagram on the Gaussian sphere. The cylinder and some beams are clustered into one class because they overlap on the Gaussian sphere. Figure 8(e) is the corresponding Clustering Effect Diagram to a spacing surface.

In contrast to DBSCAN, AQ-DBSCAN quickly selects the representative points to extend the region, and with the precondition of the same result after the algorithm calculating, the clustering speed has been greatly improved. It is clear that from the above-mentioned experimental results, shown in Figures 9, 10, and 11, the Gauss figures show that the clustering effect is identical.

The final segmentation effect is as shown in Figure 12.

Table 1 compares the time consumed by the three experiments described above, and the time unit is milliseconds. Figure 13 is an analysis of time-consumption for AQ-DBSCAN and DBSCAN, and Figure 14 is an analysis of time-consumption for AQ-DBSCAN and FDBSCAN. It can be seen from the time-consumption comparison that the AQ-DBSCAN algorithm is less time-consuming than the DBSCAN and FDBSCAN algorithms. The AQ-DBSCAN algorithm accelerates significantly as the number of points increases. For example, when 1,441,999 points are used, DBSCAN is six times more time-consuming than AQ-DBSCAN, and when 642,984 points are used, DBSCAN is 65 times more time-consuming than AQ-DBSCAN.

4. Conclusions

The segmentation method based on density clustering is one of the important methods of point cloud segmentation. The most commonly used density clustering method, the DBSCAN algorithm, can identify any shape of cluster,

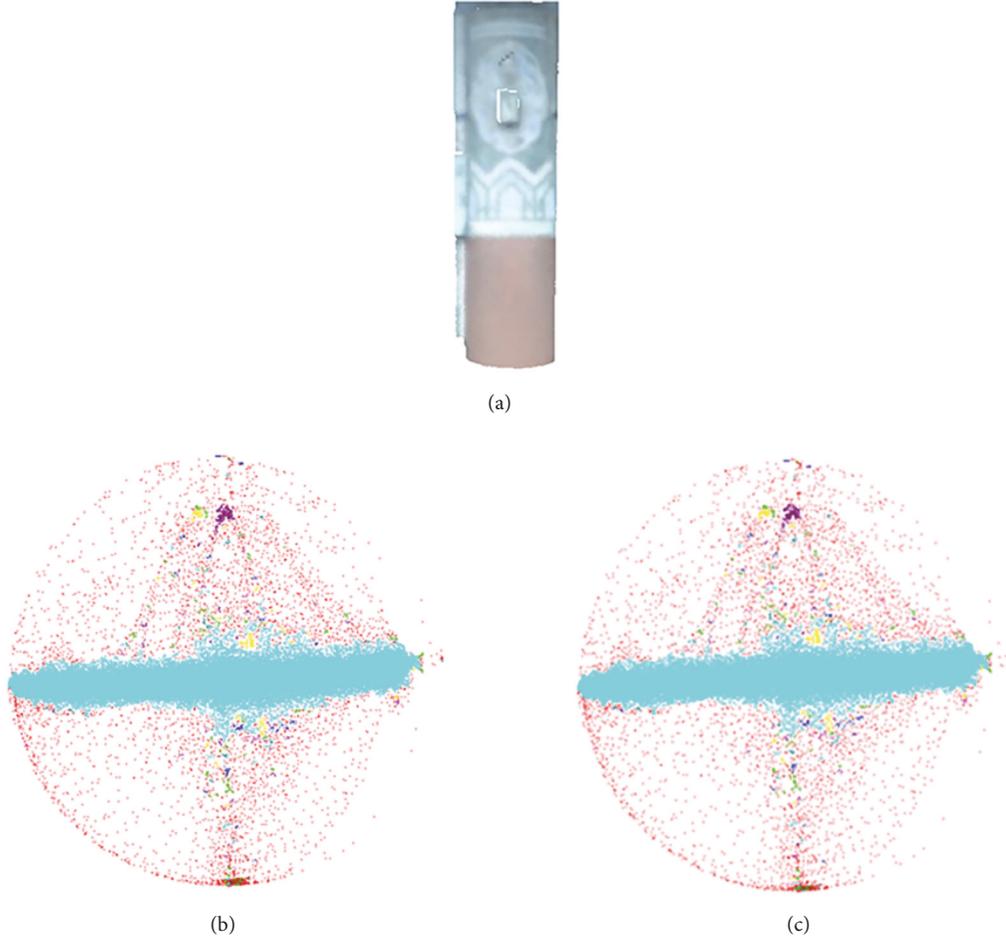


FIGURE 10: Comparison of the effects of DBSCAN clustering and AQ-DBSCAN clustering (2nd). (a) Original point cloud. (b) DBSCAN Clustering Effect Diagram. (c) AQ-DBSCAN Clustering Effect Diagram.

TABLE 1: Comparison of time consumed by DBSCAN, FDBSCAN, and AQ-DBSCAN (ms).

Figure	The size of the dataset	DBSCAN	FDBSCAN	AQ-DBSCAN
Figure 9	5771	468	46	31
Figure 10	141999	8549	1679	1607
Figure 11	642984	225608	3697	3448

and its strong applicability is demonstrated by the initial segmentation of the ancient building Gaussian map point cloud. However, there are some drawbacks to the method, such as the need for a manual definition of the parameters, as well as having low efficiency for high amounts of calculation. Aiming at addressing the disadvantages of the DBSCAN algorithm, this paper presents an improved automatic clustering algorithm, AQ-DBSCAN, which is based on the DBSCAN algorithm. The AQ-DBSCAN algorithm solves the lack of automatic estimation of the neighborhood of parameters in the DBSCAN algorithm by automatically generating the Nm curve and automatically searching for

the inflection point. In addition, this algorithm presents a reliable method of using representative points to expand the neighborhoods, thereby reducing the amount of computation, to quickly perform density clustering. Clustering experiments were performed using point cloud data collected from the Imperial Palace of the Forbidden City in Beijing, China, using a ScanStation 2 scanner. According to the comparison of DBSCAN and FDBSCAN algorithms, it is obvious that with the same clustering effect, the AQ-DBSCAN algorithm allows for automatic parameter estimation and high efficiency. Only the automatic estimation method of neighborhood radius σ is discussed in this paper.

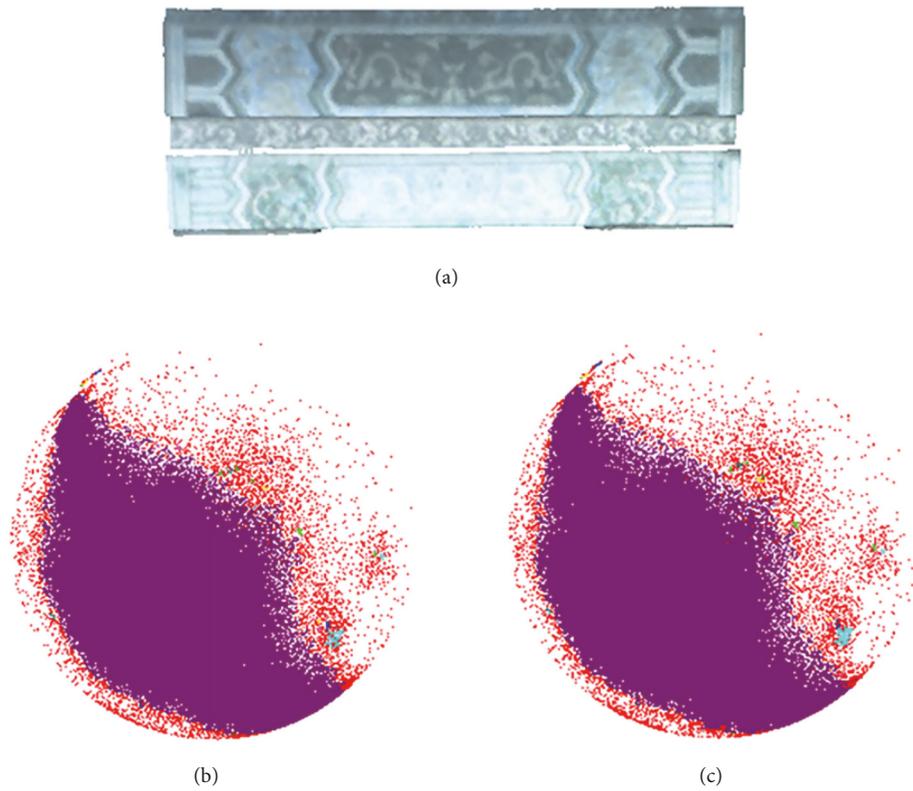


FIGURE 11: Comparison of the effects of DBSCAN clustering and AQ-DBSCAN clustering (3rd). (a) Original point cloud. (b) DBSCAN Clustering Effect Diagram. (c) AQ-DBSCAN Clustering Effect Diagram.

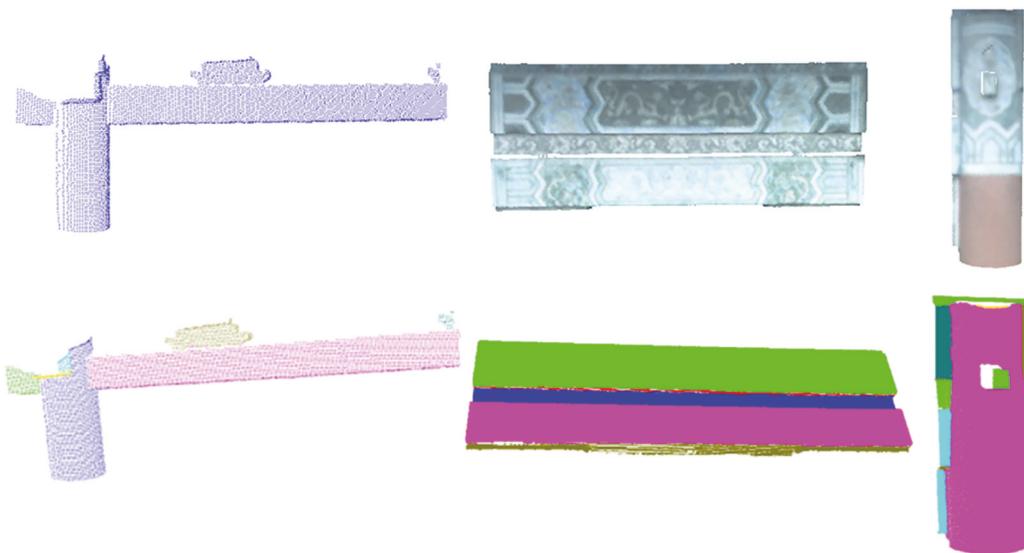


FIGURE 12: The final segmentation effect of the AQ-DBSCAN algorithm.

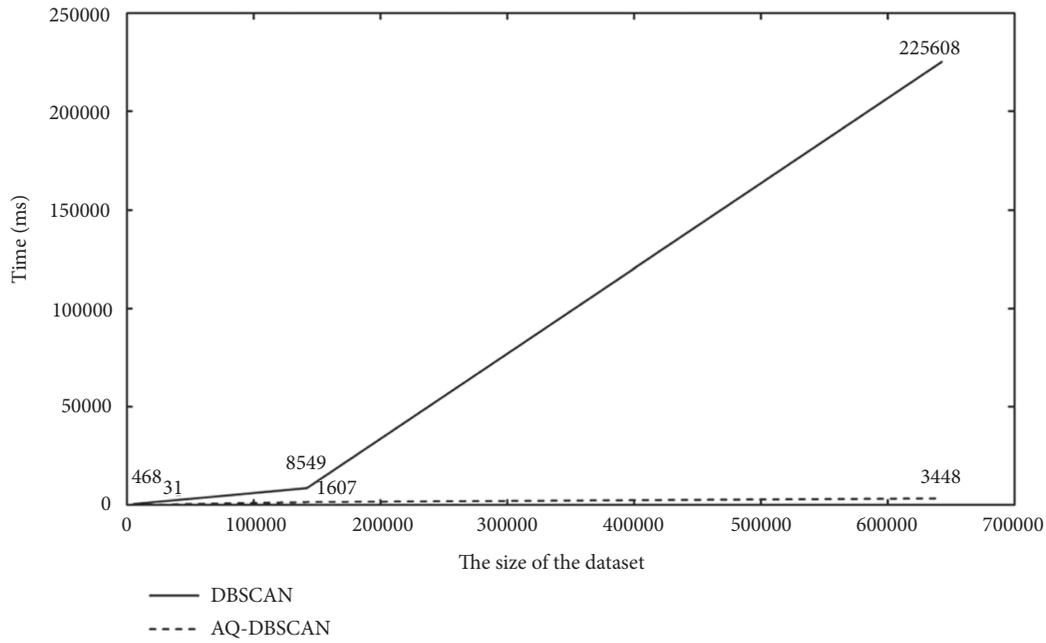


FIGURE 13: Comparison of time consumption for DBSCAN and AQ-DBSCAN.

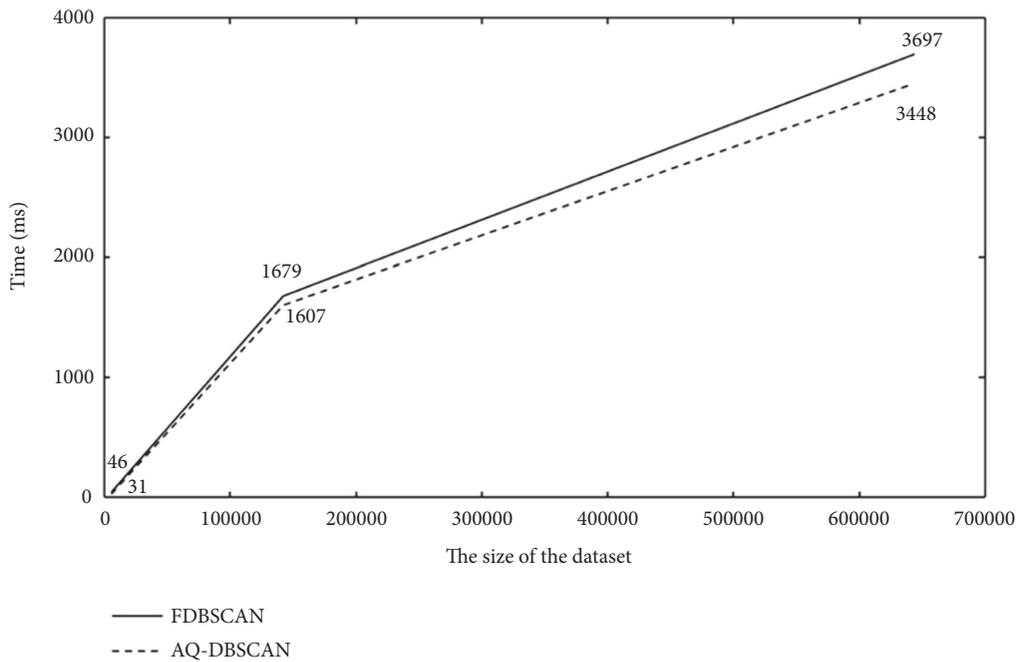


FIGURE 14: Comparison of time consumption for FDBSCAN and AQ-DBSCAN.

The other parameter, MinPts, is defined as 4 according to convention. Further work will focus on the automatic adaptive definition of MinPts based on differences in point cloud density.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (41601409 and 41301429) and the Beijing Natural Science Foundation (8172016).

References

- [1] P. J. Besl and R. C. Jain, "Segmentation through variable-order surface fitting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 2, pp. 167–192, 2002.
- [2] N. Yokoya and M. D. Levine, "Range image segmentation based on differential geometry: a hybrid approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 6, pp. 643–649, 1989.
- [3] X. Jiang, H. Bunke, and U. Meier, "High-level feature based range image segmentation," *Image and Vision Computing*, vol. 18, no. 10, pp. 817–822, 2000.
- [4] P. Shi and G. Vosselman, "Automatic extraction of building features from terrestrial laser scanning," *International Archives of the Photogrammetry Remote Sensing & Spatial Information Sciences*, vol. 36, pp. 25–27, 2006.
- [5] L. Kaufman and P. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, DBLP, 2009.
- [6] R. Schnabel, R. Wahl, and R. Klein, "Efficient RANSAC for point-cloud shape detection," *Computer Graphics Forum*, vol. 26, no. 2, pp. 214–226, 2007.
- [7] J. Zhao, Y. Wang, R. Zhang et al., "Clustering method for building laser scan point cloud data," CN 104573705A, 2015.
- [8] L. N. Xia and J. W. Jing, "Sa-dbscan: a self-adaptive density-based clustering algorithm," *Journal of the Graduate School of the Chinese Academy of Sciences*, vol. 26, no. 4, pp. 530–538, 2009.
- [9] J. Huang and C.-H. Menq, "Automatic data segmentation for geometric feature extraction from unorganized 3-D coordinate points," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp. 268–279, 2001.
- [10] P. Liang and J. S. Todhunter, "Representation and recognition of surface shapes in range images: a differential geometry approach," *Image Vision Computing*, vol. 51, no. 2, pp. 78–109, 1990.
- [11] J. Martínez, A. Soria-Medina, P. Arias, and A. F. Buffara-Antunes, "Automatic processing of Terrestrial Laser Scanning data of building façades," *Automation in Construction*, vol. 22, pp. 298–305, 2012.
- [12] T. Rabbani, F. A. V. D. Heuvel, and G. Vosselman, "Segmentation of point clouds using smoothness constraint," *International Archives of Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 36, pp. 248–253, 2006.
- [13] D. Tóvári and N. Pfeifer, "Segmentation based robust interpolation - a new approach to laser data filtering," *Proceedings of the ISPRS Workshop on Laser Scanning*, vol. 36, pp. 79–84, 2005.
- [14] J. Overby, L. Bodum, E. Kjems et al., "Automatic 3d building reconstruction from airborne laser scanning and cadastral data using hough transform," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 35, no. B3, 2004.
- [15] M. G. Vosselman, B. G. H. Gorte, G. Sithole et al., "Recognising structure in laser scanning point clouds," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 36, no. 8/W2, pp. 33–38, 2004.
- [16] F. Bretar and M. Roux, "Hybrid image segmentation using LiDAR 3D planar primitives," in *Proceedings of ISPRS - Workshop "Laser Scanning 2005"*, vol. 6, pp. 12–14, 2005.
- [17] R. Schnabel, R. Wahl, and R. Klein, "Shape detection in point clouds," Computer Graphics Technical report CG-2006-2, 2006.
- [18] R. Schnabel, R. Wahl, R. Wessel et al., "Shape recognition in 3D point clouds," Computer Graphics Technical report CG-2007-1, 2007.
- [19] S. O. Elberink and M. G. Vosselman, "Adding the third dimension to a topographic database using airborne laser scanner data," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 36, part 3, pp. 92–97, 2006.
- [20] M. Roggero, "Object segmentation with region growing and principal component analysis," *Pediatric Allergy & Immunology*, vol. 12, no. 2, pp. 289–292, 2002.
- [21] R. Gelbard, O. Goldman, and I. Spiegler, "Investigating diversity of clustering methods: an empirical comparison," *Data & Knowledge Engineering*, vol. 63, no. 1, pp. 155–166, 2007.
- [22] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, John Wiley & Sons, New York, NY, USA, 1990.
- [23] M. Ester, H. P. Kriegel, and J. Sander, "A density based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, pp. 2226–2231, 1996.
- [24] S. Zhou, A. Zhou, W. Jin et al., "Fdbscan: a fast dbscan algorithm," *Journal of Software*, vol. 15, no. 6, pp. 735–744, 2000.



Hindawi

Submit your manuscripts at
www.hindawi.com

